

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
APPLICATION FOR UNITED STATES LETTERS PATENT**

**for**

**METHOD AND SYSTEM FOR  
DEMONSTRATING PROOF OF CONCEPT**

**By:**

James D. Barnes  
14717 Grandview St.  
Overland Park, KS 66221  
Citizenship: USA

Robin D. Katzer  
1501 Wells Fargo Drive  
Olathe, KS 66062  
Citizenship: USA

Dora Potluri  
3800 Skyline Drive  
Plano, TX 75025  
Citizenship: USA

M. Jeffrey Stone  
12206 Bradshaw  
Overland Park, KS 66213  
Citizenship: USA

## **METHOD AND SYSTEM FOR DEMONSTRATING PROOF OF CONCEPT**

### **CROSS-REFERENCE TO RELATED APPLICATIONS**

**[0001]** None.

### **STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT**

**[0002]** Not applicable.

### **REFERENCE TO A MICROFICHE APPENDIX**

**[0003]** Not applicable.

### **FIELD OF THE INVENTION**

**[0004]** The present invention relates to the evaluation of products prior to purchase. More particularly, embodiments of the present invention provide for a standard method and system for determining and documenting whether a product meets the needs it is intended to fill.

### **BACKGROUND OF THE INVENTION**

**[0005]** A potential purchaser of a product may wish to evaluate the product prior to purchase to determine if the product meets its needs. The product can be subjected to a set of tests designed to simulate the functions the product might be expected to perform in actual use conditions. A decision on whether to purchase the product can then be based on the performance of the product in the tests. These tests can be referred to as test cases or use cases. Subjecting a product to a set of use cases and basing purchasing decisions on the performance of the product in the use cases may be referred to as a proof of concept.

### **SUMMARY OF THE INVENTION**

**[0006]** The present invention provides, in one embodiment, a system for demonstrating proof of concept of a project for an organization. The system includes a requirements component, a use

case component, a log component, and a reporting component. The requirements component is operable to maintain requirements of the organization for the project. The use case component is operable to maintain a plurality of use cases, each of the plurality of use cases are associated with at least one of the requirements. The log component is operable to track results of the project executing at least some of the plurality of use cases. The log component is also operable to track at least some defects of the project identified based on executing some of the plurality of use cases. The reporting component is operable to report results for at least some of the use cases.

[0007] The present disclosure, according to another embodiment, provides a method of demonstrating proof of concept of a product. The method includes describing requirements to be fulfilled by the product, and generating use cases defining test scenarios to test whether the product satisfies the requirements, each of the use cases based on at least one of the requirements. The method includes describing a relationship between each use case and an associated one of the requirements, and weighting each use case based upon a priority associated with the requirement tested by the use case.

[0008] One embodiment provides a method for demonstrating proof of concept of a product. The method includes providing a project plan component identifying at least one product to test, and describing requirements to be fulfilled by the product. The method includes generating use cases defining test scenarios to test whether the product satisfies the requirements, each of the use cases are based on at least one of the requirements. The method provides for describing a relationship between each use case and an associated one of the requirements, and testing the product using the use cases. The method also includes weighting each use case based upon a priority associated with the requirement tested by the use case.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0009] Figure 1 is a block diagram of the Proof of Concept method according to one embodiment.

[0010] Figure 2 is a block diagram of the Proof of Concept system according to one embodiment.

[0011] Figure 3 is a graphical user interface, according to one embodiment of the Proof of Concept method and system, for displaying project requirements.

[0012] Figure 4 is a graphical user interface, according to one embodiment of the Proof of Concept method and system, for displaying use cases.

[0013] Figure 5 is a graphical user interface, according to one embodiment of the Proof of Concept method and system, for displaying a use case log.

[0014] Figure 6 is a graphical user interface, according to one embodiment of the Proof of Concept method and system, for displaying a defects log.

## **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

[0015] Demonstrations of proof of concept are traditionally carried out by vendors of a product at the request of potential purchasers of the product. A potential purchaser might create a set of test cases or use cases designed to evaluate the product. Multiple use cases might be created to test various aspects of the product. Several vendors of similar products might then perform the tests specified by the use cases without prior knowledge of the contents of the tests. The potential purchaser might observe the performance of the products and might choose one vendor's product based on the observations.

[0016] While general procedures such as these might be followed in demonstrating proof of concept, a standardized process might not be used. A potential purchaser might follow different

procedures for different products in the creation of use cases and in the evaluation of vendors' performances in the use cases. In addition, a potential purchaser might not maintain adequate documentation to support the reasons for choosing one product over another.

**[0017]** In an embodiment of the method and system for demonstrating proof of concept described herein, a standard procedure is used for creating use cases, performing use cases, evaluating the results of use cases, and documenting plans for and conclusions indicated by use cases. A potential purchaser creates a project plan describing how the Proof of Concept is to be carried out. The project plan includes a set of use cases specifying the requirements the product is to meet, the tests to be performed, and the criteria for passing or failing. The potential purchaser, rather than a vendor, carries out the use cases by performing hands-on evaluations of the products that might be purchased. Results of the use cases are recorded in a use case log and in a defects log. A project report is created detailing the conclusions reached as a result of the use cases.

**[0018]** The Proof of Concept process typically begins when a need for a new product is identified. The need can be immediate or can relate to the long-term future direction of an enterprise. In the case of long-term needs, an enterprise might have a target state architecture that describes a desired future state of the enterprise. The Proof of Concept process can be applied to products intended to implement this target state architecture. A project plan can be created to describe the desired functionality and capabilities of the product under consideration, how the product might aid in reaching the target state or in meeting an immediate need, and how the product is to be evaluated.

**[0019]** The project plan can include detailed descriptions of the use cases to be used to evaluate the product. This can include descriptions of the requirements to be filled by the product, how the use cases test whether the requirements are met, the importance of each use case, and

criteria for determining whether the product passed or failed each use case. The requirements can be weighted based on their priority. Groups that might be affected by the product can provide feedback to the group conducting the Proof of Concept regarding the product's potential impact and the most appropriate methods for evaluating the product. This feedback can be used to revise the project plan.

**[0020]** Information needed to perform a Proof of Concept, such as potential vendors, appropriate products, and estimated costs can be gathered by the group that will perform the Proof of Concept or can be provided to that group by another group within the same enterprise. Potential vendors can then be sent a Request for Information (RFI), a Request for Proposals (RFP), or some other document specifying what is expected from their product. In an embodiment, some vendors can be eliminated prior to the Proof of Concept process based on their responses to the RFI or RFP. This preliminary elimination of some vendors can be conducted by the same group that performs the Proof of Concept or by another group.

**[0021]** When a need has been identified, vendors whose products might fill the need have been selected, and use cases to evaluate the selected products have been created, the vendors of the products are requested to submit samples of their products. The potential purchaser then subjects the samples to the tests specified in the use cases. In an alternative embodiment, rather than a competition among multiple vendors, the Proof of Concept process can be performed to evaluate a single vendor's product. In another alternative, the Proof of Concept process can be performed on an idea or product generated internally within an enterprise.

**[0022]** Four general types of tests can be performed in the Proof of Concept process: functional, performance, scalability, and longevity. Functional testing determines whether or not a product achieves the results it is intended to. Performance tests deal with how well a product

achieves its intended results. For example, for a software product interacting with a database, performance measurements might include lookup times, update times, lookup overhead, and message rates. Scalability testing deals with how well a product can adapt to changes in the volume of work it is requested to do. Longevity tests measure how long a product can operate without errors. As appropriate for the product under consideration, some of these tests could be eliminated or additional tests could be performed.

[0023] The results of the use case tests can be documented in a use case log, where the requirements listed in the project plan are mapped to use cases. One requirement could be mapped to one or more use cases or one use case could be mapped to one or more requirements. Notes indicating whether each use case passed or failed are maintained in the use case log. For use cases that failed, the defects that caused the failure are also listed.

[0024] The defects in the use case log are mapped to the defects log. The defects log gives detailed information on each defect such as who created the defect, when it was created, reasons for each failure, the likely impact of each failure, the level of effort that might be required to implement a workaround for the failure, the vendor's response to the defect, whether the vendor is promising a fix, and the date of the fix. In an embodiment, different levels of severity could be assigned to different defects. Also, while the use case log and the defects log have been described as separate entities, they could alternatively be a single document. As used herein, the term "document" may refer to the process or act of obtaining or capturing information, and may also refer to a printed or electronic document, file, or set of files stored as one or more hard-copy or computer files accessible singularly or as a whole by one or more means or from one ore more interfaces.

**[0025]** When the tests conducted in the use cases are complete, the potential purchaser can review the information collected and reach a decision on which tested product is most appropriate for meeting the identified need. A project report can then be created detailing the results of the use cases and the conclusions that have been reached. Potentially impacted groups, such as the future users of the products under consideration, can provide feedback on a preliminary project report and this feedback can be incorporated into a revised project report.

**[0026]** An embodiment of the Proof of Concept process is illustrated in Figure 1. It should be understood that the procedures described below do not necessarily need to occur in the order depicted in Figure 1. Similar results could be achieved with other sequences. In addition, some of the individual procedures could be omitted without detracting from the effectiveness of the overall process.

**[0027]** In box 112, a need for a new product is identified. In box 114, a preliminary project plan is created to identify a product that can fill the need. In box 116, a preliminary project plan document is published to the main architects and internal peers who are likely to be impacted by the new product. Resources for determining an appropriate product are assigned in box 118. In box 120, feedback from the architects and peers regarding the preliminary project plan is received. The preliminary project plan is updated based on this feedback in box 122. In box 124, the updated project plan is published.

**[0028]** In box 126, the requirements to be met by the new product are defined based on the project plan and the target state architecture. In box 128, use cases are developed based on the requirements, such as using earlier use cases, test result, develop new use cases, or other information maintained by the present disclosure. The use cases detail the features to be tested, the test conditions that will be used to test the product, and the pre-conditions and post-conditions for



each test. In box 130, the project plan is updated to assign use cases to resources so progress can be tracked on a regular basis.

[0029] The use cases are executed in box 132. In box 134, the passing or failing of a use case is noted in the use case log and the defect log. Defects in products or enhancements needed by products are tracked. In box 136, regular phone calls or other means of communication are set up to communicate issues with vendors.

[0030] When the use case tests are complete, a preliminary report documenting the findings and recommendations is written in box 138. In box 140, the preliminary report is published to the group that performed the use cases. In box 142, the feedback of this group is received and, in box 144, the feedback is incorporated into the preliminary report. In box 146, the project report is created with the final results of the use cases, the defects discovered, and the recommendations.

[0031] In box 148, the project report is presented to stakeholders in the new product. Feedback is received from the stakeholders in box 150 and the project report is updated based on this feedback in box 152. In box 154, a final recommendation on the product to be purchased is made. In box 156, the vendors are informed of the recommendation. A group that will implement the new product is informed of the recommendation in box 158.

[0032] Information and documentation related to the Proof of Concept process can be stored in a computer-based system such as that shown in Figure 2. The Proof of Concept system 200 consists of a computer 210 through which a user can create, read, update, or delete information in a project plan document 220, a document describing a set of use cases 230, a use case log 240, a defect log 250, and a project report 260. The document describing the use cases 230 might be a stand-alone document as shown in Figure 2 or, as described above, might be a part of the project plan document 220. The use case log 240 and the defects log 250 might be separate documents as

shown in Figure 2 or might be a single document. Other embodiments of the documents, such as the combining of the use cases 230, the use case log 240, and the defects log 250 into a single document, are also possible.

**[0033]** An example of a product requirements section of the project plan document 220 is shown in Figure 3. Column 310 lists reference numbers that map each requirement to a section of a Request for Information (RFI) document sent to potential vendors. Column 312 lists a set of requirements. In this case, the requirements deal with a system interface adapter 316, data transformation 318, and transaction fallout management 320. Under each general requirement 316, 318, and 320, several sub-requirements are listed. For example, under the system interface adapter requirement 316, the sub-requirements are protocol connectivity 322, core capability 324, custom development 326, and development 328. Under each sub-requirement 322, 324, 326, and 328, several specific requirements are listed. For example, under the protocol connectivity sub-requirement 322, the specific requirements are RDBMS 330, CORBA 332, J2EE/EJB 334, JMS 336, and MQSeries 338.

**[0034]** Each of the specific requirements listed in column 312 is mapped to a use case shown in column 314. For example, the RDBMS requirement 330 is mapped to use case SB-9 340, the CORBA requirement 332 is mapped to use case SB-8 342, the J2EE/EJB requirement 334 is mapped to use case SB-1 344, the JMS requirement 336 is mapped to use case SB-7 346, and the MQSeries requirement 338 is mapped to use case SB-6 348.

**[0035]** An example of a detailed list of use cases (230 in Figure 2) is shown in Figure 4. In column 410, the use case numbers are listed. These map to the use case numbers shown in column 314 of Figure 3. In column 412, a set of requirements is listed. These map to the sub-requirements 316, 318, and 320 shown in Figure 3. The only requirement shown in column 412 of Figure 4 is

‘system interface adapter’, which maps to the system interface adapter sub-requirement 316 in Figure 3. Column 414 lists a set of requirements that map to the specific requirements 330, 332, 334, 336, and 338 shown in Figure 3. For example, the third item in column 414 is ‘system interface adapter with CORBA and complex flow’ 418. This corresponds to the third item in column 410, which is use case number 8 420. These two items in turn map to CORBA 332 and SB-8 342, respectively, in the requirements document shown in Figure 3.

**[0036]** Column 416 lists the specific use cases intended to test the requirements shown in column 414. In the case of the ‘system interface adapter with CORBA and complex flow’ requirement 418, the use cases are interface definition 422, synchronous invocation interface 424, Java/RMI interface 426, can invoke external system interface adapter 428, data transformation 430, CORBA interface with the system interface adapter 432, and support for flow specification 434.

**[0037]** An example of a use case log (240 in Figure 2) is shown in Figure 5. Column 510 of the log 500 lists a series of use cases. These use cases map to the use cases shown in column 410 of Figure 4 and in column 314 in Figure 3. For example, use case SB-8 518 corresponds to use case 8 420 in Figure 4 and to use case SB-8 342 in Figure 3. Column 512 lists the titles of the corresponding use cases in column 510. For example, the title corresponding to use case SB-8 518 is SIA – CORBA SIA 520. These titles map to the requirements in column 414 in Figure 4 and to the requirements in column 312 in Figure 3.

**[0038]** Column 514 lists the pass or fail status of each use case. For example, the status of use case SB-8 518 is ‘fail’ 522. Column 516 lists defect numbers for each use case that has a ‘fail’ status. These are code numbers that cross-reference detailed explanations of each defect. For example, use case SB-8 518 has a defect code 524 of 5. A reference document can be consulted to learn more about the defect referred to by this defect code.

**[0039]** Figure 6 shows an example of a detailed defects log (250 in Figure 2). A use case 612 is listed that maps to use case SB-8 342 in Figure 3, use case 8 420 in Figure 4, and use case SB-8 518 in Figure 5. A brief description section 614 describes the functionality to be tested by the use case 612. A requirement cross reference table 616 lists other requirements related to the use case 612. An importance section 618 describes the relevance of the use case 612. A status section 620 states whether the use case 612 passed or failed. A reasons for failure section 622 gives detailed explanations of the causes of a failed use case 612. A workarounds section 624 describes potential fixes for a failed use case 612. A level of effort section 626 describes the amount of work that might be needed to implement a workaround 624.

**[0040]** Documents such as those depicted in Figures 3, 4, 5, and 6 can be incorporated into a project report (260 in Figure 2) that summarizes the use cases to which a product has been subjected, the results of the use cases, and the conclusions that have been reached based on the use case results.

**[0041]** In an alternative embodiment, information and documentation related to the Proof of Concept process can be incorporated into a web site. For example, the project plan, the list of use cases, the use case log, the defects log, and the project report could be published as web pages that can be viewed and updated by users of the Proof of Concept process.

**[0042]** It should be appreciated that a number of additional benefits may be derived by having access to the variety of use cases and test results maintained by the present disclosure. For example, in a system employing a target state architecture or targeted architecture standard, a number of requirements are likely to reoccur with some frequency since standardized components are employed throughout the enterprise. Thus, when one product is tested by various use cases, the next product to be tested may likely be able to reuse a number of the previous use cases, due to, for

example, standardization of system interfaces throughout the enterprise. Further, the product may not need to be tested against a given project interfaces where the product had been previously test for the same interface for another project. Maintaining the benchmarks and/or results from the product tests and use cases creates efficiencies, since some potential products may be readily selected or eliminated based on results of previous product testing by use cases for similar systems and interfaces.

**[0043]** Thus, the present disclosure, according to one embodiment provides ready access to reusable use cases, as well as test results and/or benchmarks of the product performing the use cases. Also, the present disclosure provides a synergy to target state architecture enterprises by leveraging cross-project product identification, rejection and/or selection, based on information derived from the present disclosure.

**[0044]** Although only a few embodiments of the present invention have been described, it should be understood that the present invention may be embodied in many other specific forms without departing from the spirit or the scope of the present invention. The present examples are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.